

Cell Connect

Legacy SMS Message Support

Application Note

A-CELL

Document No. D117-013

Document Revision 1.0

06/2016

CONTENTS

1. Preface	2
1.1. Purpose of this Document.....	2
1.2. Additional Information.....	2
1.3. Support.....	2
2. Application Description.....	3
3. Setup	4
3.1. Logix Message Instruction.....	4
4. Operation	7
4.1. Logix Message Instruction.....	7
5. Notes.....	8



1. PREFACE

1.1. PURPOSE OF THIS DOCUMENT

This document will assist the user to setup Legacy SMS Logix message instructions (previously supported by the 1756HP-CELL module) using the Cell Connect Module.

1.2. ADDITIONAL INFORMATION

The following resources contain additional information that can assist the user with the module installation and operation.

Resource	Link
Slate Installation	http://www.aparian.com/software/slate
User Manual, Datasheet Example Code & UDTs	http://www.aparian.com/products/cellconnect
Ethernet wiring standard	www.cisco.com/c/en/us/td/docs/video/cds/cde/cde205_220_420/installation/guide/cde205_220_420_hig/Connectors.html

Table 1.1. - Additional Information

1.3. SUPPORT

Technical support will be provided via the Web (in the form of user manuals, FAQ, datasheets etc.) to assist with installation, operation, and diagnostics.

For additional support the user can use either of the following:

Contact Us web link	www.aparian.com/contact-us
Support email	support@aparian.com

Table 1.2. - Additional Support

2. APPLICATION DESCRIPTION

The Aparian Cell Connect module can be used as a replacement for the SMS message functionality supported by the discontinued 1756HP-CELL module. The Cell Connect module implements a message instruction almost identical to that used by the legacy 1756HP-CELL which allows the sending of SMS messages from the Logix environment.

This functionality allows the user to dynamically configure and send SMS messages triggered by the Logix application code.

3. SETUP

The following section will describe the configuration and implementation of the Logix message instruction required to send SMS messages (using the legacy message instruction method).

3.1. LOGIX MESSAGE INSTRUCTION

Below is an example of the message instruction used to send a text message from the Logix environment.

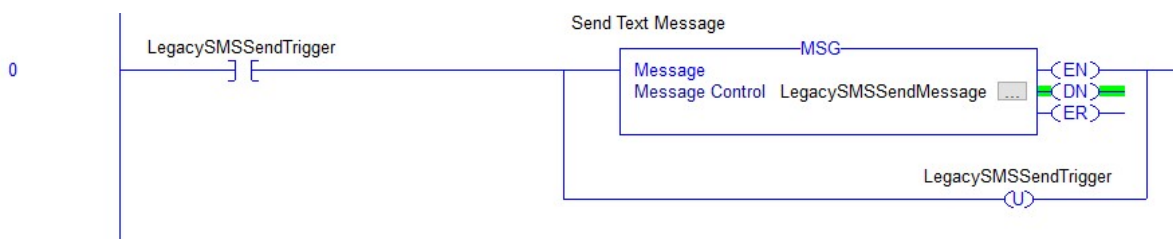


Figure 3.1. – Send Text Message Logix

Figure 3.2. – Message Instruction Parameters

Parameter	Value / Description
Message Type	CIP Generic
Service Type	Custom
Service Code	79 Hex (Legacy Send Text Message service - SMS)
Class	412 Hex
Instance	1
Attribute	1
Source Element	Tag of type <i>LegacySMSSend</i>
Source Length	184
Destination Element	DINT (reserved)

Table 3.1. – Message Instruction Parameters

The structure of the source element (*LegacySMSSend*) UDT is as follows:

Name:

Description:

Members: Data Type Size: 184 byte(s)

	Name	Data Type	Style	Description	External Access
	Number	SINT[20]	ASCII		Read/Write
+	Message	AparianString160			Read/Write
100 010					

Figure 3.3. – Text Parameters source structure

The structure of the text message element in the Source UDT (*AparianString160*) UDT is as follows:

Name:

Description:

Maximum Characters:

Members: Data Type Size: 164 byte(s)

Name	Data Type	Style	Description	External Access
LEN	DINT	Decimal		Read/Write
DATA	SINT[160]	ASCII		Read/Write

Figure 3.4. – Text parameters sub structure.

4. OPERATION

The following section will describe the operation of the Logix message instruction required to send SMS messages (using the legacy message instruction method).

4.1. LOGIX MESSAGE INSTRUCTION

The user must populate two parameters to successfully send a text message.

The first parameter is the Number. The number is a 20-byte array (as shown below) which is formatted as ASCII text. In the example below the number *0712345678* will be texted.

The next parameter to be updated is the actual Text Message to be sent. This is in the format of a 160 character string and can be updated in Logix similar to any other string. In the example below the text message read *"This is a test message"*.

LegacySMSSend	{...}	{...}		LegacySMSSend
LegacySMSSend.Number	{...}	{...}	ASCII	SINT[20]
LegacySMSSend.Number[0]	'0'		ASCII	SINT
LegacySMSSend.Number[1]	'7'		ASCII	SINT
LegacySMSSend.Number[2]	'1'		ASCII	SINT
LegacySMSSend.Number[3]	'2'		ASCII	SINT
LegacySMSSend.Number[4]	'3'		ASCII	SINT
LegacySMSSend.Number[5]	'4'		ASCII	SINT
LegacySMSSend.Number[6]	'5'		ASCII	SINT
LegacySMSSend.Number[7]	'6'		ASCII	SINT
LegacySMSSend.Number[8]	'7'		ASCII	SINT
LegacySMSSend.Number[9]	'8'		ASCII	SINT
LegacySMSSend.Number[10]	'\$00'		ASCII	SINT
LegacySMSSend.Number[11]	'\$00'		ASCII	SINT
LegacySMSSend.Number[12]	'\$00'		ASCII	SINT
LegacySMSSend.Number[13]	'\$00'		ASCII	SINT
LegacySMSSend.Number[14]	'\$00'		ASCII	SINT
LegacySMSSend.Number[15]	'\$00'		ASCII	SINT
LegacySMSSend.Number[16]	'\$00'		ASCII	SINT
LegacySMSSend.Number[17]	'\$00'		ASCII	SINT
LegacySMSSend.Number[18]	'\$00'		ASCII	SINT
LegacySMSSend.Number[19]	'\$00'		ASCII	SINT
LegacySMSSend.Message	'This is a test message'	{...}		AparianString160

Figure 4.1. – Example values for Message Source Structure.

Once the message has been successfully sent the Logix Message instruction will set the *Done* flag (in the example used it is the LegacySMSSendMessage.DN bit). If an error occurred while trying to send the SMS the Logix Message Instruction will set the *Error* flag (in the example used it is the LegacySMSSendMessage.ER bit).

5. NOTES

- The user must remember to update the communication path of the Logix Message Instruction to the new target (Cell Connect Module).
- The service code and class has changed from the message supported by the 1756HP-CELL so the user must ensure that this is updated.
- The user will first need to download to the module before the unconnected message instruction will work.
- Refer to the supplied example code for a working solution of using a Legacy SMS Message Instruction.